



Docket No.: 61282-064

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Customer Number: 20277
Minoru OKAMOTO : Confirmation Number:
Serial No.: 10/787,166 : Group Art Unit:
Filed: February 27, 2004 : Examiner:
For: SYSTEM LSI DESIGN SUPPORT APPARATUS AND A SYSTEM LSI
DESIGN SUPPORT METHOD

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

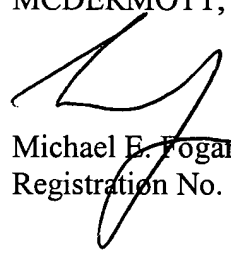
At the time the above application was filed, priority was claimed based on the following application:

Japanese Patent Application No. 2003-051046, filed February 27, 2003

A copy of each priority application listed above is enclosed.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:prg
Facsimile: (202) 756-8087
Date: May 27, 2004

日本国特許庁
JAPAN PATENT OFFICE

61282-064
Okamoto
Feb. 27, '04
10787,166

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 2月27日
Date of Application:

出願番号 特願2003-051046
Application Number:
[ST. 10/C]: [JP 2003-051046]

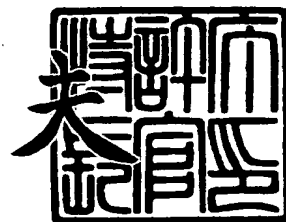
出願人 松下電器産業株式会社
Applicant(s):



特許庁長官
Commissioner,
Japan Patent Office

2003年12月12日

今井 康



出証番号 出証特2003-3103507

【書類名】 特許願

【整理番号】 5037940152

【提出日】 平成15年 2月27日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/50

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 岡本 稔

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100105647

【弁理士】

【氏名又は名称】 小栗 昌平

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100105474

【弁理士】

【氏名又は名称】 本多 弘徳

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100108589

【弁理士】

【氏名又は名称】 市川 利光

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100115107

【弁理士】

【氏名又は名称】 高松 猛

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100090343

【弁理士】

【氏名又は名称】 栗宇 百合子

【電話番号】 03-5561-3990

【手数料の表示】

【予納台帳番号】 092740

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0002926

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 システム L S I 設計支援装置及び設計支援方法

【特許請求の範囲】

【請求項 1】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置であって、

システムの機能を高級言語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、

前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文の数を計数する解析手段と、

前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を出力する集計手段と、
を備えたことを特徴とするシステム L S I 設計支援装置。

【請求項 2】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文のネスティング数を計数すること
を特徴とする請求項 1 に記載のシステム L S I 設計支援装置。

【請求項 3】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする請求項 1
又は 2 に記載のシステム L S I 設計支援装置。

【請求項 4】 処理ユニットが処理するに好適な条件分岐文の数、条件分岐文のネスティング数及び条件分岐文の条件に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする請求項 1 から 3 のいずれか一項に記載のシステム L S I 設計支援装置。

【請求項 5】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置であって、

システムの機能を高級言語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、

前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文の数を計数する解析手段と、

前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を入力する集計手段と、
を備えたことを特徴とするシステム L S I 設計支援装置。

【請求項 6】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文のネスティング数を計数することを特徴とする請求項 5 に記載のシステム L S I 設計支援装置。

【請求項 7】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする請求項 5 又は 6 に記載のシステム L S I 設計支援装置。

【請求項 8】 処理ユニットが処理するに好適なループ制御文の数、ループ制御文のネスティング数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする請求項 5 から 7 のいずれか一項に記載のシステム L S I 設計支援装置。

【請求項 9】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置であって、

システムの機能を高級言語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、

前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の数を計数する解析手段と、

前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を入力する集計手段と、

を備えたことを特徴とするシステム L S I 設計支援装置。

【請求項 10】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文のネスティング数を計数することを特徴とする請求項 9 に記載のシステム L S I 設計支援装置。

【請求項 11】 前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする請求項 9 又は 10 に記載のシステム L S I 設計支援装置。

【請求項 12】 処理ユニットが処理するに好適な条件分岐文及びループ制御文の数、条件分岐文及びループ制御文のネスティング数並びに条件分岐文及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする請求項 9 から 11 のいずれか一項に記載のシステム L S I 設計支援装置。

【請求項 13】 前記関数抽出手段が抽出した関数から任意に選択した複数の関数の組を少なくとも 1 つ指定する関数組み合わせ手段を備え、前記解析手段は、前記関数組み合わせ手段が指定した関数の組それぞれについて解析を行うことを特徴とする請求項 1 から 12 のいずれか一項に記載のシステム L S I 設計支援装置。

【請求項 14】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援方法であって、

システムの機能を高級言語で記述したプログラムを入力するステップと、

前記プログラムに記述された関数を抽出するステップと、

抽出したそれぞれの関数について、その内部に記述された条件分岐文の数を計数するステップと、

前記計数結果をそれぞれの関数について集計した集計結果情報を生成するステ

ップと、

を含むことを特徴とするシステム L S I 設計支援方法。

【請求項 15】 前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文のネスティング数を計数するステップを含むことを特徴とする請求項 14 に記載のシステム L S I 設計支援方法。

【請求項 16】 前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを含むことを特徴とする請求項 14 又は 15 に記載のシステム L S I 設計支援方法。

【請求項 17】 処理ユニットが処理するに好適な条件分岐文の数、条件分岐文のネスティング数及び条件分岐文の条件に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、

前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップと、

を含むことを特徴とする請求項 14 から 16 のいずれか一項に記載のシステム L S I 設計支援方法。

【請求項 18】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援方法であって、

システムの機能を高級言語で記述したプログラムを入力するステップと、

前記プログラムに記述された関数を抽出するステップと、

抽出されたそれぞれの関数について、その内部に記述されたループ制御文の数を計数するステップと、

前記計数結果をそれぞれの関数について集計した集計結果情報を生成するステップと、

を含むことを特徴とするシステム L S I 設計支援方法。

【請求項 19】 前記抽出されたそれぞれの関数について、その内部に記述されたループ制御文のネスティング数を計数するステップを含むことを特徴とす

る請求項 18 に記載のシステム L S I 設計支援方法。

【請求項 20】 前記抽出されたそれぞれの関数について、その内部に記述されたループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを含むことを特徴とする請求項 18 又は 19 に記載のシステム L S I 設計支援方法。

【請求項 21】 処理ユニットが処理するに好適なループ制御文の数、ループ制御文のネスティング数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、

前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップと、

を含むことを特徴とする請求項 18 から 20 のいずれか一項に記載のシステム L S I 設計支援方法。

【請求項 22】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援方法であって、

システムの機能を高級言語で記述したプログラムを入力するステップと、

前記プログラムに記述された関数を抽出するステップと、

抽出されたそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の数を計数するステップと、

前記計数結果をそれぞれの関数について集計した集計結果情報を生成するステップと、

を含むことを特徴とするシステム L S I 設計支援方法。

【請求項 23】 前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文のネスティング数を計数するステップを含むことを特徴とする請求項 22 に記載のシステム L S I 設計支援方法。

【請求項 24】 前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数及びループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを

含むことを特徴とする請求項 22 又は 23 に記載のシステム LSI 設計支援方法。

【請求項 25】 処理ユニットが処理するに好適な条件分岐文及びループ制御文の数、条件分岐文及びループ制御文のネスティング数並びに条件分岐文の条件に係る変数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、

前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップと、

を含むことを特徴とする請求項 22 から 24 のいずれか一項に記載のシステム LSI 設計支援方法。

【請求項 26】 前記抽出した関数から任意に選択した複数の関数の組を少なくとも 1 つ指定するステップを含み、前記指定された関数の組それぞれについて前記集計を行うことを特徴とする請求項 14 から 25 のいずれか一項に記載のシステム LSI 設計支援方法。

【請求項 27】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム LSI にシステムの機能を割り当てる設計を支援するシステム LSI 設計支援装置に提供するデータを格納するデータベース装置であって、

前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適な条件分岐文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とするデータベース装置。

【請求項 28】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム LSI にシステムの機能を割り当てる設計を支援するシステム LSI 設計支援装置に提供するデータを格納するデータベース装置であって、

前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適なループ制御文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とするデータベース装置。

【請求項 29】 複数の異なったアーキテクチャをもつ処理ユニットからなるシステム LSI にシステムの機能を割り当てる設計を支援するシステム LSI

設計支援装置に提供するデータを格納するデータベース装置であって、

前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適な条件分岐文及びループ制御文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とするデータベース装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、高級言語で記述されたシステムのプログラムが実現する機能を、マイクロプロセッサ、デジタル信号処理プロセッサ、専用ロジック回路等から構成されるシステム用のLSIに割り付けることを支援するシステムLSI設計支援装置及び設計支援方法に関するものである。

【0002】

【従来の技術】

近年、LSIの高集積化に伴い多数の機能からなるシステムを1チップのLSIに集積することが可能となっている。このLSI（以下、システムLSIと称する）は、異なる種類の処理ユニット（例えば、マイクロプロセッサ、デジタル信号処理プロセッサ、専用のロジック回路）及びメモリより構成され、システムの様々な機能がそれぞれ適切な処理ユニットで実行される。

【0003】

処理ユニットは、それぞれ異なったアーキテクチャを持っており、得意とする処理内容がある。例えば、マイクロプロセッサ（以下、CPUと称する）ではLSI外部から入力される多様な制御情報の判断や、複雑なアルゴリズムをもつ処理を、高級言語を元にしたプログラムで実行するのに適する。

【0004】

また、デジタル信号処理プロセッサ（以下、DSPと称する）は複雑な数値演算を高速で実行するのに向いている。さらに、専用ロジック回路は単純な数値演算等を高効率で実行できるが、一方で複雑な判断条件を伴った処理には向かないという特徴がある。

【0005】

システムの機能をどの処理ユニットに割り当てるかは以下の過程でなされる。まず、システムの動作の一部あるいは全体を記述したプログラムを元に簡略な解析を行う。この解析でプログラムの機能単位（即ちモジュール、C言語等の高級言語では関数）毎の処理時間が見積もられる。この情報を基に設計者は、システムの機能を適切な処理ユニットに割り付ける。次に、処理ユニット毎にさらに詳しく処理時間や実行サイクル数を算出する。この段階で問題がなければ、さらに処理ユニットに応じた組み込みプログラム、例えばCPUで実行するプログラムやDSP用のプログラムを作成し、同時に専用ロジック回路を設計することで正確な処理時間を算出する。

【0006】

以上の煩雑な一連の過程を支援する設計支援方法が提案されている（例えば、特許文献1を参照）。特許文献1によれば、複数の関数から構成される高級言語で記述されたプログラムをコンパイルして得られるオブジェクトコードを入力してシミュレーションを実行し、予め指定した区間について処理時間と処理回数を記録、出力することで処理時間の大小に応じてハードウェア化すべき箇所、すなわち専用ロジックで処理すべき箇所を選択している。

【0007】

【特許文献1】

特開平9-160949号公報（第4-14頁、第1図）

【0008】

【発明が解決しようとする課題】

しかしながら、上記従来の方法では、処理時間を基準にしてシステムのある機能を専用ロジックに割り付けることは支援できるものの、それ以外の基準では何ら判断できる情報を提供できないという問題点がある。すなわち、例えばシステムが有する一部の機能で、単純な数値演算を多用する機能Aを実現するに当たり、CPUで処理時間が満足されれば、専用ロジックで処理する要請はなくなる。ところが、この場合CPUにとっては不得意な処理を実行することになり、不必要な動作を多分に実行することにより消費電力の増加を招く。

【0009】

また別の例として、システムが有する一部の機能で、条件判断を多用した複雑なアルゴリズムを持つ機能Bを実現するに当たり、予め決められた時間内にCPUが処理できなければ、専用ロジックに機能が割り当てられることになる。ところが、このような複雑な処理を専用ロジックで実現しようとする多段の状態遷移制御回路が必要となり、専用ロジックの回路規模が増加して、同時にその設計期間は膨大となる。

【0010】

本発明は上記事情に鑑みてなされたもので、システムLSIに内蔵されている異なったアーキテクチャを持つ処理ユニットに、システムが担う機能を適切に割り当てられるように支援するシステムLSI設計支援装置及び設計支援方法を提供することを目的とする。

【0011】

【課題を解決するための手段】

上記問題点を解決するために請求項1に記載のシステムLSI設計支援装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステムLSIにシステムの機能を割り当てる設計を支援するシステムLSI設計支援装置であって、システムの機能を高級言語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文の数を計数する解析手段と、前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を出力する集計手段とを備えたことを特徴とする。

【0012】

請求項14に記載のLSI設計支援方法は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステムLSIにシステムの機能を割り当てる設計を支援するシステムLSI設計支援方法であって、システムの機能を高級言語で記述したプログラムを入力するステップと、前記プログラムに記述された関数を抽出するステップと、抽出したそれぞれの関数について、その内部に記述された条件分岐文の数を計数するステップと、前記計数結果をそれぞれの関数について集

計した集計結果情報を生成するステップとを含むことを特徴とする。

【0013】

上記構成によれば、条件分岐文の数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0014】

請求項 2 に記載のシステム L S I 設計支援装置は、請求項 1 に記載のシステム L S I 設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文のネスティング数を計数することを特徴とする。

【0015】

請求項 15 に記載の L S I 設計支援方法は、請求項 14 に記載の L S I 設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文のネスティング数を計数するステップを含むことを特徴とする。

【0016】

上記構成によれば、条件分岐文のネスティング数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0017】

請求項 3 に記載のシステム L S I 設計支援装置は、請求項 1 又は 2 に記載のシステム L S I 設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする。

【0018】

請求項 16 に記載の L S I 設計支援方法は、請求項 14 又は 15 に記載のシステム L S I 設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを含むことを特徴とする。

【0019】

上記構成によれば、条件分岐文の条件に係る変数を生成するに要した関数の数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0020】

請求項 4 に記載のシステム L S I 設計支援装置は、請求項 1 から 3 のいずれか一項に記載のシステム L S I 設計支援装置において、処理ユニットが処理するに好適な条件分岐文の数、条件分岐文のネスティング数及び条件分岐文の条件に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする。

【0021】

請求項 17 に記載の L S I 設計支援方法は、請求項 14 から 16 のいずれか一項に記載のシステム L S I 設計支援方法において、処理ユニットが処理するに好適な条件分岐文の数、条件分岐文のネスティング数及び条件分岐文の条件に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップとを含むことを特徴とする。

【0022】

上記構成によれば、条件分岐文の数、条件分岐文のネスティング数、条件分岐文の条件に係る変数の生成に要した関数の数から選択した複数の組み合わせに関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0023】

請求項 5 に記載のシステム L S I 設計支援装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置であって、システムの機能を高級言

語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文の数を計数する解析手段と、前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を出力する集計手段とを備えたことを特徴とする。

【0024】

請求項18に記載のLSI設計支援方法は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステムLSIにシステムの機能を割り当てる設計を支援するシステムLSI設計支援方法であって、システムの機能を高級言語で記述したプログラムを入力するステップと、前記プログラムに記述された関数を抽出するステップと、抽出されたそれぞれの関数について、その内部に記述されたループ制御文の数を計数するステップと、前記計数結果をそれぞれの関数について集計した集計結果情報を生成するステップとを含むことを特徴とする。

【0025】

上記構成によれば、ループ制御文の数に関するシステムLSI設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0026】

請求項6に記載のシステムLSI設計支援装置は、請求項5に記載のシステムLSI設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文のネスティング数を計数することを特徴とする。

【0027】

請求項19に記載のLSI設計支援方法は、請求項18に記載のシステムLSI設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述されたループ制御文のネスティング数を計数するステップを含むことを特徴とする。

【0028】

上記構成によれば、ループ制御文のネスティング数に関するシステムLSI設

計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0029】

請求項7に記載のシステムLSI設計支援装置は、請求項5又は6に記載のシステムLSI設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述されたループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする。

【0030】

請求項20に記載のLSI設計支援方法は、請求項18又は19に記載のシステムLSI設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述されたループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを含むことを特徴とする。

【0031】

上記構成によれば、ループ制御文の繰り返し回数に係る変数を生成するに要した関数の数に関するシステムLSI設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0032】

請求項8に記載のシステムLSI設計支援装置は、請求項5から7のいずれか一項に記載のシステムLSI設計支援装置において、処理ユニットが処理するに好適なループ制御文の数、ループ制御文のネスティング数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする。

【0033】

請求項21に記載のLSI設計支援方法は、請求項18から20のいずれか一

項に記載のシステム L S I 設計支援方法において、処理ユニットが処理するに好適なループ制御文の数、ループ制御文のネスティング数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップとを含むことを特徴とする。

【0034】

上記構成によれば、ループ制御文の数、ループ制御文のネスティング数、ループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせに関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0035】

請求項 9 に記載のシステム L S I 設計支援装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置であって、システムの機能を高級言語で記述したプログラムを入力とし、前記プログラムに記述された関数を抽出する関数抽出手段と、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の数を計数する解析手段と、前記解析手段の計数結果を前記関数抽出手段で抽出されたそれぞれの関数について集計した集計結果情報を出力する集計手段とを備えたことを特徴とする。

【0036】

請求項 22 に記載の L S I 設計支援方法は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援方法であって、システムの機能を高級言語で記述したプログラムを入力するステップと、前記プログラムに記述された関数を抽出するステップと、抽出されたそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の数を計数するステップと、前記計数結果をそれぞれの関数について集計した集計結果情報を生成するステップとを含むことを特徴

とする。

【0037】

上記構成によれば、条件分岐文及びループ制御文の数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0038】

請求項 10 に記載のシステム L S I 設計支援装置は、請求項 9 に記載のシステム L S I 設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文のネスティング数を計数することを特徴とする。

【0039】

請求項 23 に記載の L S I 設計支援方法は、請求項 22 に記載のシステム L S I 設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文のネスティング数を計数するステップを含むことを特徴とする。

【0040】

上記構成によれば、条件分岐文及びループ制御文のネスティング数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0041】

請求項 11 に記載のシステム L S I 設計支援装置は、請求項 9 又は 10 に記載のシステム L S I 設計支援装置において、前記解析手段は、前記関数抽出手段が抽出したそれぞれの関数について、その内部に記述された条件分岐文及びループ制御文の繰り返し回数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数することを特徴とする。

【0042】

請求項 24 に記載の L S I 設計支援方法は、請求項 22 又は 23 に記載のシステム L S I 設計支援方法において、前記抽出されたそれぞれの関数について、その内部に記述された条件分岐文の条件に係る変数及びループ制御文の繰り返し回

数に係る変数の計算過程に基づいて、前記変数の生成に要した関数の数を計数するステップを含むことを特徴とする。

【0043】

上記構成によれば、条件分岐文の条件に係る変数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数に関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0044】

請求項 12 に記載のシステム L S I 設計支援装置は、請求項 9 から 11 のいずれか一項に記載のシステム L S I 設計支援装置において、処理ユニットが処理するに好適な条件分岐文及びループ制御文の数、条件分岐文及びループ制御文のネスティング数並びに条件分岐文及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報と前記集計手段が出力する集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるマッピング手段を備えることを特徴とする。

【0045】

請求項 25 に記載の L S I 設計支援方法は、請求項 22 から 24 のいずれか一項に記載のシステム L S I 設計支援方法において、処理ユニットが処理するに好適な条件分岐文及びループ制御文の数、条件分岐文及びループ制御文のネスティング数並びに条件分岐文の条件に係る変数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせを異なる処理ユニット毎に予め定義した定義情報を入力するステップと、前記定義情報と前記集計結果情報とを照合して適切な処理ユニットを関数毎に対応付けるステップとを含むことを特徴とする。

【0046】

上記構成によれば、条件分岐文及びループ制御文の数、条件分岐文及びループ制御文のネスティング数並びに条件分岐文の条件に係る変数及びループ制御文の繰り返し回数に係る変数の生成に要した関数の数から選択した複数の組み合わせに関するシステム L S I 設計支援情報に基づいて異なったアーキテクチャを持つ

処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0047】

請求項13に記載のシステムLSI設計支援装置は、請求項1から12のいずれか一項に記載のシステムLSI設計支援装置において、前記関数抽出手段が抽出した関数から任意に選択した複数の関数の組を少なくとも1つ指定する関数組み合わせ手段を備え、前記解析手段は、前記関数組み合わせ手段が指定した関数の組それぞれについて解析を行うことを特徴とする。

【0048】

請求項26に記載のLSI設計支援方法は、請求項14から25のいずれか一項に記載のシステムLSI設計支援方法において、前記抽出した関数から任意に選択した複数の関数の組を少なくとも1つ指定するステップを含み、前記指定された関数の組それぞれについて前記集計を行うことを特徴とする。

【0049】

上記構成によれば、関連性の高い関数を同じ処理ユニットで処理する場合についてのシステムLSI設計支援情報に基づいて異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てることができる。

【0050】

請求項27に記載のデータベース装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステムLSIにシステムの機能を割り当てる設計を支援するシステムLSI設計支援装置に提供するデータを格納するデータベース装置であって、前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適な条件分岐文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とする。

【0051】

上記構成によれば、条件分岐文に関するシステムLSI設計支援情報を生成するためのデータをシステムLSI設計支援装置に提供することができる。

【0052】

請求項28に記載のデータベース装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステムLSIにシステムの機能を割り当てる設計を支

援するシステム L S I 設計支援装置に提供するデータを格納するデータベース装置であって、前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適なループ制御文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とする。

【0053】

上記構成によれば、ループ制御文に関するシステム L S I 設計支援情報を生成するためのデータをシステム L S I 設計支援装置に提供することができる。

【0054】

請求項 29 に記載のデータベース装置は、複数の異なったアーキテクチャをもつ処理ユニットからなるシステム L S I にシステムの機能を割り当てる設計を支援するシステム L S I 設計支援装置に提供するデータを格納するデータベース装置であって、前記データは、システムの機能を高級言語で記述したプログラムに関し、処理ユニットが処理するに好適な条件分岐文及びループ制御文の数を、異なる処理ユニット毎に予め定義した定義情報であることを特徴とする。

【0055】

上記構成によれば、条件分岐文及びループ制御文に関するシステム L S I 設計支援情報を生成するためのデータをシステム L S I 設計支援装置に提供することができる。

【0056】

【発明の実施の形態】

以下、本発明の実施の形態について図面を参照して説明する。本発明に係るシステム L S I 設計支援装置を用いて設計を行うシステム L S I は、異なったアーキテクチャを持つ複数の処理ユニット（CPU、DSP、専用ロジック）で構成されている（システム L S I の構成は図示略）。

【0057】

（実施の形態 1）

図 1 は、本発明の実施の形態 1 におけるシステム L S I 設計支援装置の構成を示すブロック図である。図 1 において、コンピュータ 100 は、プロセッサ 110 及び主記憶装置 120 を備える。プロセッサ 110 は、各種演算処理やシステ

ム L S I 設計支援装置全体の統括制御処理を行う。主記憶装置 120 は、本発明のシステム L S I 設計支援装置の動作に関わる手続きを記述したプログラムやデータなどを格納する。また、補助記憶装置 130 は、コンピュータ 100 に入力するデータを、補助記憶装置 140 は、コンピュータ 100 から出力されるデータを格納する。

【0058】

システムの機能の一部あるいは全部を高級言語で記述したプログラム 132 は、補助記憶装置 130 に予め格納されている（格納手段は図示略）。図 2 は、実施の形態 1 において補助記憶装置に格納されるプログラム 132 の記述例である。図に示すように、プログラムは C 言語で記述され、複数の関数で構成されている。

【0059】

次に、上記構成のシステム L S I 設計支援装置の動作について説明する。プログラム 132 がコンピュータ 100 に入力されると、主記憶装置 120 の関数抽出部 122、解析処理部 124、集計処理部 126、の順番で各種処理が実行される。

【0060】

関数抽出部 122 は、プログラム 132 に記述されている関数を抽出する。プログラム 132 が図 2 に示した内容の場合、関数として `func__A`、`func__B`、`func__C` が抽出される。解析処理部 124 は、抽出された関数ごとに条件分岐文の数を計数する。

【0061】

図 3 は、実施の形態 1 における関数内部の解析処理の流れを示すフローチャートである。関数抽出部 122 より抽出された関数の一つを選択する（ステップ S101）。プログラム 132 が図 2 に示す内容である場合、まず `func__A` が選択される。次に、その関数内に含まれる条件分岐文の数をカウントする（ステップ S102）。図 2 の `func__A` の内部では、条件分岐文として `if` 文を使用しており、その数は `if` 文 211、`if` 文 212、`if` 文 213 の合計 3 個であるので、その数を記録する（ステップ S103）。以上の処理がすべての関数について

終了したかどうかを判定し（ステップS104）、終了していない場合はステップS101からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0062】

集計処理部126は、解析処理部124で得られた条件分岐文の数を集計し、集計結果を生成する。図4は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0063】

実施の形態1の構成により、システムの機能を記述したプログラム132の関数毎の特徴が得られる。例えば、上記のプログラム例ではfunc__Bで条件分岐文が10箇所使用されているため、条件判断を効率よく行うCPUでこの関数を処理するのが適切である、というような設計支援情報が取得できる。

【0064】

（実施の形態2）

本発明の実施の形態2におけるシステムLSI設計支援装置について、図1、図2、図5及び図6を参照しながら説明する。

【0065】

先の実施の形態と異なる点は、図1の解析処理部124において、抽出された関数ごとに条件分岐文のネスティング数までを計数することにある。

【0066】

図5は、実施の形態2における関数内部の解析処理の流れを示すフローチャートである。解析処理部124は、関数抽出部122より抽出された関数の一つを選択する（ステップS201）。プログラム132が図2に示す内容である場合、まずfunc__Aが選択される。次に、その関数内に含まれる条件分岐文の数をカウントする（ステップS202）。図2では、3個のif文がカウントされる。更にこの関数内に含まれる条件分岐文のネスティング数（入れ子構造の段数）と、対象となる条件分岐文の数をカウントする（ステップS203）。図2のfunc__Aの中にあるif文212は、if文211の内部で1段のネスティング構造となっている。従って、このときネスティング1段の条件分岐文は1とカウン

トし、記録する（ステップS204）。以上の処理がすべての関数について終了したかどうかを判定し（ステップS205）、終了していない場合はステップS201からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0067】

集計処理部126は、解析処理部124で得られた条件分岐文の数とネスティング数を集計し、集計結果を生成する。図6は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0068】

実施の形態2の構成により、実施の形態1とは異なる別の観点から、プログラム132の関数毎の特徴が得られる。一般に、条件分岐のネスティング段数が多い程処理が複雑となるため、専用ロジックでの実行は困難になる。例えば、上記のプログラム例ではfunc__Aでネスティングを用いた条件分岐文が使用されているため、条件判断を効率よく行うCPUでこの関数进行处理するのが適切である、というような設計支援情報が取得できる。

【0069】

（実施の形態3）

本発明の実施の形態3におけるシステムLSI設計支援装置について、図1、図7、図8及び図9を参照しながら説明する。

【0070】

先の実施の形態と異なる点は、図1の解析処理部124において、抽出された関数ごとに、条件分岐文の条件を生成する過程を追跡して、条件変数を生成するに要した関数の数を計数し、内部構文の解析を行うことにある。図7は、実施の形態3において補助記憶装置に格納されるプログラム132の記述例である。

【0071】

図8は、実施の形態3における関数内部の解析処理の流れを示すフローチャートである。解析処理部124は、関数抽出部122より抽出された関数の一つを選択する（ステップS301）。プログラム132が図7に示す内容である場合

、まず `func__A` が選択される。次に、その関数内に含まれる条件分岐文で用いられる条件に係る変数を生成するに当たりいくつかの関数を経由するかを解析する（ステップ S302）。図 7 の条件分岐文 211 で用いられている変数 `AIN1` は `func__A` 外部の関数で計算されており、変数 `AIN2` は外部から定数として与えられるものとする。図 7 に示されるように、変数 `AIN1` は、`func__D` 内部で計算され、`func__D` 内部において 2 つの文 711, 712 に記述の `func__A1`、`func__A2` により変数 `AIN1` が生成されている。従って、`AIN1` を生成するに要する関数は 2 となる。また、変数 `AIN2` は外部から定数で与えられているので生成に必要な関数は 0 である。このような解析に基づいて、条件変数を生成するに要した関数の数と変数の数を記録する（ステップ S303）。以上の処理がすべての関数について終了したかどうかを判定し（ステップ S304）、終了していない場合はステップ S301 からの処理を再び繰り返し、終了した場合は次の集計処理部 126 へ移行する。

【0072】

集計処理部 126 は、解析処理部 124 で得られた条件分岐文の数と条件変数を生成するに要した関数の数と変数の数を集計し、集計結果を生成する。図 9 は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報 142 は、補助記憶装置 140 に格納される。

【0073】

実施の形態 3 の構成により、実施の形態 1、2 とは異なる別の観点から、プログラム 132 の関数毎の特徴が得られる。一般に、条件分岐で用いられる変数を生成するに当たりその手順が多い程処理が複雑となるため、専用ロジックでの実行は困難になる。例えば、上記のプログラム例では `func__A` の条件分岐文で用いる変数を生成するには 2 個の関数が必要であるため、条件判断を効率よく行う CPU でこの関数を処理するのが適切である、というような設計支援情報が取得できる。

【0074】

（実施の形態 4）

本発明の実施の形態 4 におけるシステム LSI 設計支援装置について、図 1、

図10、図11及び図12を参照しながら説明する。

【0075】

先の実施の形態と異なる点は、図1の解析処理部124において、抽出された関数ごとにループ制御文の数を計数することにある。図10は、実施の形態4において補助記憶装置に格納されるプログラム132の記述例である。

【0076】

図11は、実施の形態4における関数内部の解析処理の流れを示すフローチャートである。解析処理部124は、関数抽出部122より抽出された関数の一つを選択する（ステップS401）。プログラム132が図10に示す内容である場合、まずfunc__LAが選択される。次に、その関数内に含まれるループ制御文の数をカウントする（ステップS402）。図10のfunc__LAの内部では、ループ制御文としてfor文を使用しており、その数はfor文1011、for文1012、for文1013の合計3個であるので、その数を記録する（ステップS403）。以上の処理がすべての関数について終了したかどうかを判定し（ステップS404）、終了していない場合はステップS401からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0077】

集計処理部126は、解析処理部124で得られたループ制御文の数を集計し、集計結果を生成する。図12は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0078】

実施の形態4の構成により、システムの機能を記述したプログラム132の関数毎の特徴が得られる。例えば、上記のプログラム例ではfunc__LBでループ制御文が10箇所使用されているため、ループ制御を効率よく行うDSP又は専用ロジックでこの関数を処理するのが適切である、というような設計支援情報が取得できる。

【0079】

（実施の形態5）

本発明の実施の形態5におけるシステムLSI設計支援装置について、図1、

図10、図13及び図14を参照しながら説明する。

【0080】

先の実施の形態と異なる点は、図1の解析処理部124において、抽出された関数ごとにループ制御文のネスティング数までを計数することにある。

【0081】

図13は、実施の形態5における関数内部の解析処理の流れを示すフローチャートである。解析処理部124は、関数抽出部122より抽出された関数の一つを選択する（ステップS501）。プログラム132が図10に示す内容である場合、まずfunc__LAが選択される。次に、その関数内に含まれるループ制御文の数をカウントする（ステップS502）。図10では、3個のfor文がカウントされる。更にこの関数内に含まれるループ制御文のネスティング数（入れ子構造の段数）と、対象となるループ制御文の数をカウントする（ステップS503）。図10のfunc__LAの中にあるfor文1012は、for文1011の内部で1段のネスティング構造となっている。従って、このときネスティング1段のループ制御文は1とカウントし、記録する（ステップS504）。以上の処理がすべての関数について終了したかどうかを判定し（ステップS505）、終了していない場合はステップS501からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0082】

集計処理部126は、解析処理部124で得られたループ制御文の数とネスティング数を集計し、集計結果を生成する。図14は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0083】

実施の形態5の構成により、実施の形態4とは異なる別の観点から、プログラム132の関数毎の特徴が得られる。一般に、ループ制御文のネスティング段数が多い程処理が複雑となるため、専用ロジックでの実行は困難になる。例えば、上記のプログラム例ではfunc__LAでネスティングを用いたループ制御文が使用されているため、システムLSIに内蔵されるDSPでこの関数を処理する

のが適切である、というような設計支援情報が取得できる。

【0084】

(実施の形態6)

本発明の実施の形態6におけるシステムLSI設計支援装置について、図1、図15、図16及び図17を参照しながら説明する。

【0085】

先の実施の形態と異なる点は、図1の解析処理部124において、抽出された関数ごとに、ループ制御文の繰り返し回数を生成する過程を追跡して、条件変数を生成するに要した関数の数を計数し、内部構文の解析を行うことにある。図15は、実施の形態6において補助記憶装置に格納されるプログラム132の記述例である。

【0086】

図16は、実施の形態6における関数内部の解析処理の流れを示すフローチャートである。解析処理部124は、関数抽出部122より抽出された関数の一つを選択する(ステップS601)。プログラム132が図15に示す内容である場合、まずfunc_LAが選択される。次に、ループ制御文で用いられる繰り返し回数に係る変数を生成するに当たりいくつかの関数を経由するかを解析する(ステップS602)。図15のループ制御文1011で用いられている繰り返し回数を指示する変数AIN1は、func_LD内部で計算され、func_LD内部において2つの文1511、1512に記述のfunc_LA1、func_LA2により変数AIN1が生成されている。従って、AIN1を生成するに要する関数は2となる。このような解析に基づいて、条件変数を生成するに要した関数の数を記録する(ステップS603)。以上の処理がすべての関数について終了したかどうかを判定し(ステップS604)、終了していない場合はステップS601からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0087】

集計処理部126は、解析処理部124で得られたループ制御文の数と条件変数を生成するに要した関数の数と変数の数を集計し、集計結果を生成する。図1

7 は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報 142 は、補助記憶装置 140 に格納される。

【0088】

実施の形態 6 の構成により、実施の形態 5 とは異なる別の観点から、プログラム 132 の関数毎の特徴が得られる。一般にループ制御で用いられる繰り返し回数を指示する変数を生成するに当たりその手順が多い程処理が複雑となるため、専用ロジックでの実行は困難になる。例えば、上記のプログラム例では `func__LA` のループ制御文で用いる変数を生成するには 2 個の関数が必要であるため、システム LSI に内蔵される DSP でこの関数を処理するのが適切である、というような設計支援情報が取得できる。

【0089】

(実施の形態 7)

本発明の実施の形態 7 におけるシステム LSI 設計支援装置について、図 1、図 18 及び図 19 を参照しながら説明する。

【0090】

本実施の形態では、図 1 の解析処理部 124 において、抽出された関数ごとに、条件分岐文の数及びループ制御文の数、条件分岐文のネスティング数及びループ制御文のネスティング数、をすべて計数し、さらに、条件分岐文の条件及びループ制御文の繰り返し回数を生成する過程を追跡して、条件分岐文の変数を生成するに要した関数の数及びループ制御文の変数を生成するに要した関数の数を計数して、内部構文の解析を行う。尚、本実施の形態におけるプログラム 132 は、図 2、図 7、図 10、図 15 に示す内容をすべて含むプログラムであるものとする。

【0091】

図 18 は、実施の形態 7 における関数内部の解析処理の流れを示すフローチャートである。解析処理部 124 は、関数抽出部 122 より抽出された関数の一つを選択する（ステップ S701）。次に、その関数内に含まれる条件分岐文の数をカウントし（ステップ S702）、その条件分岐文の数を記録する（ステップ S703）。次に、この関数内に含まれる条件分岐文のネスティング数（入れ子

構造の段数)と対象となる条件分岐文の数をカウントし(ステップS704)、そのネスティング数を記録する(ステップS705)。次に、この関数内に含まれる条件分岐文で用いられる条件に係る変数を生成するに当たりいくつかの関数を経由するかを追跡し(ステップS706)、この条件変数を生成するに要した関数の数を記録する(ステップS707)。

【0092】

続いて、その関数内に含まれるループ制御文の数をカウントし(ステップS708)、そのループ制御文の数を記録する(ステップS709)。次に、この関数内に含まれるループ制御文のネスティング数(入れ子構造の段数)と対象となるループ制御文の数をカウントし(ステップS710)、そのネスティング数を記録する(ステップS711)。次に、この関数内に含まれるループ制御文で用いられる繰り返し回数に係る変数を生成するに当たりいくつかの関数を経由するかを追跡し(ステップS712)、この条件変数を生成するに要した関数の数を記録する(ステップS713)。以上の処理がすべての関数について終了したかどうかを判定し(ステップS714)、終了していない場合はステップS701からの処理を再び繰り返し、終了した場合は次の集計処理部126へ移行する。

【0093】

集計処理部126は、解析処理部124で得られた条件分岐文の数及びループ制御文の数、条件分岐文のネスティング数及びループ制御文のネスティング数、条件分岐文の変数を生成するに要した関数の数及びループ制御文の変数を生成するに要した関数の数、をそれぞれ集計して集計結果を生成する。図19は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0094】

実施の形態7の構成により、前述の実施の形態1から6までを総合した観点から、プログラム132の関数毎の特徴が得られ、総合的な設計支援情報が取得できる。

【0095】

(実施の形態8)

本発明の実施の形態 8 におけるシステム L S I 設計支援装置について、図 7、図 8、図 20 及び図 21 を参照しながら説明する。

【0096】

先の実施の形態と異なる点は、抽出した関数ごとに条件分岐文の条件を生成する過程を追跡するのではなく、複数の関数からなる関数の「組」について解析を行うことにある。図 20 は、本発明の実施の形態 8 におけるシステム L S I 設計支援装置の構成を示すブロック図である。補助記憶装置 130 には、プログラム 132 に加えファイル 2032 が格納されている。ファイル 2032 には任意の関数を複数選択した関数の「組」のリストがあり、関数抽出部 2122 は、プログラム 132 に記述された関数を抽出した後、ファイル 2032 に指定された「組」のリストを参照して特定の関数を選択する。以降の解析処理や集計処理は、個々の関数ではなく、この「組」に対して実行される。

【0097】

上記構成のシステム L S I 設計支援装置の動作について説明する。尚、本実施の形態におけるプログラム 132 は、図 7 で示す内容を含むプログラムとして説明する。

【0098】

まず、関数抽出部 2122 がプログラム 132 に記述されている関数を抽出し、次に、ファイル 2032 を参照する。このファイル 2032 には、プログラム 132 の中で関連性が高いと思われる複数の関数が「組」として、設計者により予め指定されている。例えば、図 7 のプログラムの `func__A` と、`func__A` で使用される変数 `AIN1` を生成する `func__D` とは関数の関連性が高いため、同一の処理ユニットで処理することが望ましい。このような場合には、`func__A` と `func__D` を、解析する関数の「組」としてファイル 2032 に指定しておけば、2 つの関数が同時に選択され、以降の解析処理が 2 つの関数に対して実行される。

【0099】

解析処理部 124 は、図 8 のフローチャートと同様の流れで解析処理を行うが、ここでは、`func__A` と `func__D` の 2 つの関数（即ち、指定した「組」

）について、条件分岐文の数と条件変数を生成するに要した関数の数を計数する。

【0100】

集計処理部126は、解析処理部124で得られた条件分岐文の数と条件変数を生成するに要した関数の数を集計し、集計結果を生成する。図21は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0101】

上記のように、予め指定した関数の「組」に対して条件分岐文に関する解析処理を実行するため、例えば、互いに関連性の高い関数func__Aとfunc__Dとを別々の処理ユニットで処理する場合についての解析が予め排除され、従ってシステムLSIの設計に係る適切な設計支援情報を取得できる。

【0102】

(実施の形態9)

本発明の実施の形態9におけるシステムLSI設計支援装置について、図15、図16、図20及び図22を参照しながら説明する。

【0103】

先の実施の形態と異なる点は、抽出された関数ごとにループ制御文の繰り返し回数を生成する過程を追跡するのではなく、複数の関数からなる関数の「組」について解析を行うことにある。また、図20の解析処理部124において行う解析処理が、条件分岐文に係る処理ではなく、ループ制御文に係る処理であることにある。

【0104】

上記構成のシステムLSI設計支援装置の動作について説明する。尚、本実施の形態におけるプログラム132は、図15で示す内容を含むプログラムとして説明する。

【0105】

まず、関数抽出部2122がプログラム132に記述されている関数を抽出し、次に、ファイル2032を参照する。このファイル2032には、プログラム

132の中で関連性が高いと思われる複数の関数が「組」として、設計者により予め指定されている。例えば、図15のプログラムのfunc__LAと、func__LAで使用される変数AIN1を生成するfunc__LDとは関連性が高いため、同一の処理ユニットで処理することが望ましい。このような場合には、func__LAとfunc__LDを、解析する関数の「組」としてファイル2032に指定しておけば、2つの関数が同時に選択され、以降の解析処理が2つの関数に対して実行される。

【0106】

解析処理部124は、図16のフローチャートと同様の流れで解析処理を行うが、ここでは、func__LAとfunc__LDの2つの関数（即ち、指定した「組」）について、ループ制御文の数と繰り返し回数を生成するに要した関数の数を計数する。

【0107】

集計処理部126は、解析処理部124で得られたループ制御文の数と繰り返し回数を生成するに要した関数の数を集計し、集計結果を生成する。図22は、集計処理結果例を概念的に表で示した模式図である。この集計結果情報142は、補助記憶装置140に格納される。

【0108】

上記のように、予め指定した関数の「組」に対してループ制御文に関する解析処理部124を実行するため、例えば、互いに関連性の高い関数func__LAとfunc__LDとを別々の処理ユニットで処理する場合についての解析が予め排除され、従ってシステムLSIの設計に係る適切な設計支援情報が取得できる。

【0109】

（実施の形態10）

本発明の実施の形態10におけるシステムLSI設計支援装置について、図2、図3、図4、図23、図24及び図25を参照しながら説明する。

【0110】

先の実施の形態と異なる点は、処理ユニットを割り当てるための関数に係る条

件を記述した情報に基づいて、関数毎に適切な処理ユニットを割り当てるためのマッピング処理部 2328 が、主記憶装置 120 の中に付加されていることにある。図 23 は、本発明の実施の形態 10 におけるシステム LSI 設計支援装置の構成を示すブロック図である。補助記憶装置 130 には、プログラム 132 に加え処理ユニットを割り当てるための関数に係る条件を記述した情報（即ち、ライブラリ）2332 が格納されている。図 24 は、処理ユニットを割り当てるための条件分岐文に関する条件例を概念的に表で示した模式図である。

【0111】

上記構成のシステム LSI 設計支援装置の動作について説明する。尚、本実施の形態におけるプログラム 132 は、図 2 で示す内容を含むプログラムとして説明する。

【0112】

まず、関数抽出部 122 は、プログラム 132 に記述されている関数を抽出する。次に、解析処理部 124 は、図 3 のフローチャートと同様の流れで、抽出された関数ごとに条件分岐文の数を計数する。さらに、集計処理部 126 は、解析処理部 124 で得られた条件分岐文の数を集計し、図 4 に示す集計結果を生成する。

【0113】

マッピング処理部 2328 は、得られた集計結果に基づいて、ライブラリ 2332 を参照して適切な処理ユニットを関数ごとに割り当てる。例えば、図 4 に示すように func_A は条件分岐文の数が 3 であり、ライブラリ 2332（図 24）を参照すると、条件分岐文の数 3 が該当するのは DSP となる。つまり、func_A は DSP で処理することが適切であるという設計支援情報が取得できる。図 25 は、マッピング処理結果例を概念的に表で示した模式図である。このように、プログラム 132 で使用されている関数ごとに適切な処理ユニットが割り当てられる。このマッピング処理結果情報 2342 は、補助記憶装置 140 に格納される。

【0114】

上記のように、条件分岐文を処理する観点から、プログラム 132 に含まれる

関数ごとに適切な処理ユニットを割り当てることにより、システム L S I の設計に係る適切な設計支援情報が取得できる。

【0 1 1 5】

(実施の形態 1 1)

本発明の実施の形態 1 1 におけるシステム L S I 設計支援装置について、図 1 0、図 1 1、図 1 2、図 2 3、図 2 6 及び図 2 7 を参照しながら説明する。

【0 1 1 6】

先の実施の形態と異なる点は、処理ユニットを割り当てるための関数に係る条件を記述した情報に基づいて、関数ごとに適切な処理ユニットを割り当てるためのマッピング処理部 2 3 2 8 が、主記憶装置 1 2 0 の中に付加されている点である。また、実施の形態 1 0 と異なるのは、ライブラリ 2 3 3 2 が、条件分岐文の数に関する条件ではなく、ループ制御文の数に関する条件であることにある。図 2 6 は、処理ユニットを割り当てるためのループ制御文に関する条件例を概念的に表で示した模式図である。

【0 1 1 7】

上記構成のシステム L S I 設計支援装置の動作について説明する。尚、本実施の形態におけるプログラム 1 3 2 は、図 1 0 で示す内容を含むプログラムとして説明する。

【0 1 1 8】

まず、関数抽出部 1 2 2 は、プログラム 1 3 2 に記述されている関数を抽出する。次に、解析処理部 1 2 4 は、図 1 1 のフローチャートと同様の流れで、抽出された関数ごとにループ制御文の数を計数する。さらに、集計処理部 1 2 6 は、解析処理部 1 2 4 で得られたループ制御文の数を集計し、図 1 2 に示す集計結果を生成する。

【0 1 1 9】

マッピング処理部 2 3 2 8 は、得られた集計結果に基づいて、ライブラリ 2 3 3 2 を参照して適切な処理ユニットを関数ごとに割り当てる。例えば、図 1 2 に示すように f u n c _ B はループ制御文の数が 1 0 であり、ライブラリ 2 3 3 2 (図 2 6) を参照すると、ループ制御文の数 1 0 が該当するのは D S P となる。

つまり、func__BはDSPで処理することが適切であるという設計支援情報が取得できる。図27は、マッピング処理結果例を概念的に表で示した模式図である。このように、プログラム132で使用されている関数ごとに適切な処理ユニットが割り当てられる。このマッピング処理結果情報2342は、補助記憶装置140に格納される。

【0120】

上記のように、ループ制御文を処理する観点から、プログラム132に含まれる関数ごとに適切な処理ユニットを割り当てることにより、システムLSIの設計に係る適切な設計支援情報が取得できる。

【0121】

(実施の形態12)

本発明の実施の形態12におけるシステムLSI設計支援装置について、図18、図19、図23、図28、図29及び図30を参照しながら説明する。

【0122】

先の実施の形態と異なる点は、図23のライブラリ2332に、条件分岐文やループ制御文の数に関する条件だけではなく、条件分岐文やループ制御文のネスティング数、条件変数の生成に要する関数の数に関する条件までが含まれていることにある。図28は、処理ユニットを割り当てるための条件例を概念的に表で示した模式図である。

【0123】

上記構成のシステムLSI設計支援装置の動作について説明する。尚、本実施の形態におけるプログラム132は、図2、図7、図10、図15に示す内容をすべて含むプログラムとして説明する。

【0124】

まず、関数抽出部122は、プログラム132に記述されている関数を抽出する。次に、解析処理部124は、図18のフローチャートと同様の流れで、抽出された関数ごとに、条件分岐文の数及びループ制御文の数、条件分岐文のネスティング数及びループ制御文のネスティング数、条件分岐文の変数を生成するに要した関数の数及びループ制御文の変数を生成するに要した関数の数、をそれぞれ

計数する。さらに、集計処理部 126 は、解析処理部 124 で得られた計数結果を集計し、図 19 に示す集計結果を生成する。

【0125】

マッピング処理部 2328 は、得られた集計結果に基づいて、ライブラリ 2332 を参照して適切な処理ユニットを関数ごとに割り当てる。

【0126】

ところで、関数には条件分岐文やループ制御文が混在することがあり、そのような関数についての割り当てでは、異なる処理ユニットが適切であると判定される場合が生じる。例えば、ある関数内に条件分岐文が 5 あり、ネスティング数が 1 段の条件分岐文が 1 ある場合、好ましい処理ユニットは DSP となるが、同じ関数内にループ制御文が 4 ある場合、これに対して好ましい処理ユニットは CPU となる。

【0127】

そこで、本実施の形態では、処理ユニットを割り当てるための条件（図 28）の項目ごとに重み付き係数を付与し、最も点数の高い処理ユニットが最適な処理ユニットであると判定するようにしている。

【0128】

図 29 は、処理ユニットを割り当てるための条件の項目と、重み付き係数の関係を概念的に表で示した模式図である。この情報もライブラリ 2332 に含まれる。以下、func_A について、処理ユニットの判定の仕方について説明する。図 29 中の K は各項目に割り当てられた重み付き係数である。図 29 に示されるように、func_A については、図 29 中の丸をつけた項目が該当する。該当する項目毎に重み付き係数を加算していくと、CPU は、条件分岐文のネスティング数について $1 + 2 = 3$ 、変数生成について $1 + 3 = 4$ 、ループ制御文のネスティング数について $1 + 1 = 2$ 、繰り返し回数の変数生成について $1 + 1 = 2$ となり、CPU の係数は $3 + 4 + 2 + 2 = 11$ となる。DSP は、条件分岐文の数について 6、条件分岐文のネスティング数について $2 + 2 = 4$ 、変数生成について $2 + 2 = 4$ 、ループ制御文の繰り返し回数の変数生成について $1 + 2 + 3 = 6$ となり、DSP の係数は $6 + 4 + 4 + 6 = 20$ となる。専用ロジックは、条件

分岐文のネスティング数について1、ループ制御文の数について6、ネスティング数について2、繰り返し回数の変数生成について2となり、専用ロジックの係数は $1 + 6 + 2 + 2 = 11$ となる。

【0129】

従って、処理ユニットごとに係数を比較すると、CPU=11、DSP=20、専用ロジック=11、となり、DSPが最適な処理ユニットであると判定される。その他の関数についても、同様の方法で処理ユニットにマッピングすることが可能であり。図30は、マッピング処理結果例を概念的に表で示した模式図である。

【0130】

このように、プログラム132で使用されている関数ごとに適切な処理ユニットが割り当てられる。このマッピング処理結果情報2342は、補助記憶装置140に格納される。

【0131】

上記のように、条件分岐文及びループ制御文の詳細な使用形態を基準として、プログラム132に含まれる関数ごとに適切な処理ユニットを割り当てることにより、システムLSIの設計に係る適切な設計支援情報が取得できる。

【0132】

尚、上記の実施の形態において、補助記憶装置130内に格納するライブラリ2332の情報は、図24、図26、図28に示した数値に限定されるものではなく、設計者が任意に設定してよい。また、数値が更新可能な構成としてもよい。

【0133】

また、上記の実施の形態において、条件分岐文としてC言語におけるif文を使用したか、これは他の条件分岐文、例えばswitch文等でもよい。同様に、ループ制御文としてC言語におけるfor文を使用したか、これは他のループ制御文、例えばwhile文等でもよい。

【0134】

また、上記の実施の形態において、補助記憶装置130及び140を、コンピ

ユータ 100 の外部に配置して説明したが、内部に配置した構成としても構わない。さらに、補助記憶装置 130 内に格納する情報のうち、特にライブラリ 2332 だけをデータベースとして別途独立して格納し、電気通信回線を介して接続することにより、複数のコンピュータからライブラリ 2332 の情報へアクセスが可能な構成としてもかまわない。

【0135】

また、本発明のシステム LSI 設計支援装置の機能を記述したプログラムは、上記実施の形態の記述例に限定されるものではないことは言うまでもない。

【0136】

【発明の効果】

以上説明したように、本発明によれば、システムの機能の一部あるいは全部を C 言語等の高級言語で記述したプログラムに含まれる関数毎に、条件分岐命令の数やループ制御命令の数を計数し、集計することで、システム LSI に内蔵される処理ユニットにシステムが持つ機能を適切に割り当てるための設計支援情報を得ることができる。

【図面の簡単な説明】

【図 1】

本発明の実施の形態 1 におけるシステム LSI 設計支援装置の構成を示すブロック図である。

【図 2】

実施の形態 1 において補助記憶装置に格納されるプログラムの記述例である。

【図 3】

実施の形態 1 における関数内部の解析処理の流れを示すフローチャートである。

【図 4】

集計処理結果例を概念的に表で示した模式図である。

【図 5】

実施の形態 2 における関数内部の解析処理の流れを示すフローチャートである。

【図 6】

集計処理結果例を概念的に表で示した模式図である。

【図 7】

実施の形態 3 において補助記憶装置に格納されるプログラムの記述例である。

【図 8】

実施の形態 3 における関数内部の解析処理の流れを示すフローチャートである。

【図 9】

集計処理結果例を概念的に表で示した模式図である。

【図 10】

実施の形態 4 において補助記憶装置に格納されるプログラムの記述例である。

【図 11】

実施の形態 4 における関数内部の解析処理の流れを示すフローチャートである。

【図 12】

集計処理結果例を概念的に表で示した模式図である。

【図 13】

実施の形態 5 における関数内部の解析処理の流れを示すフローチャートである。

【図 14】

集計処理結果例を概念的に表で示した模式図である。

【図 15】

実施の形態 6 において補助記憶装置に格納されるプログラムの記述例である。

【図 16】

実施の形態 6 における関数内部の解析処理の流れを示すフローチャートである。

【図 17】

集計処理結果例を概念的に表で示した模式図である。

【図 18】

実施の形態 7 における関数内部の解析処理の流れを示すフローチャートである。

【図 19】

集計処理結果例を概念的に表で示した模式図である。

【図 20】

本発明の実施の形態 8 におけるシステム L S I 設計支援装置の構成を示すブロック図である。

【図 21】

集計処理結果例を概念的に表で示した模式図である。

【図 22】

集計処理結果例を概念的に表で示した模式図である。

【図 23】

本発明の実施の形態 10 におけるシステム L S I 設計支援装置の構成を示すブロック図である。

【図 24】

処理ユニットを割り当てるための条件分岐文に関する条件例を概念的に表で示した模式図である。

【図 25】

マッピング処理結果例を概念的に表で示した模式図である。

【図 26】

処理ユニットを割り当てるためのループ制御文に関する条件例を概念的に表で示した模式図である。

【図 27】

マッピング処理結果例を概念的に表で示した模式図である。

【図 28】

処理ユニットを割り当てるための条件例を概念的に表で示した模式図である。

【図 29】

処理ユニットを割り当てるための条件の項目と、重み付き係数の関係を概念的に表で示した模式図である。

【図 3 0】

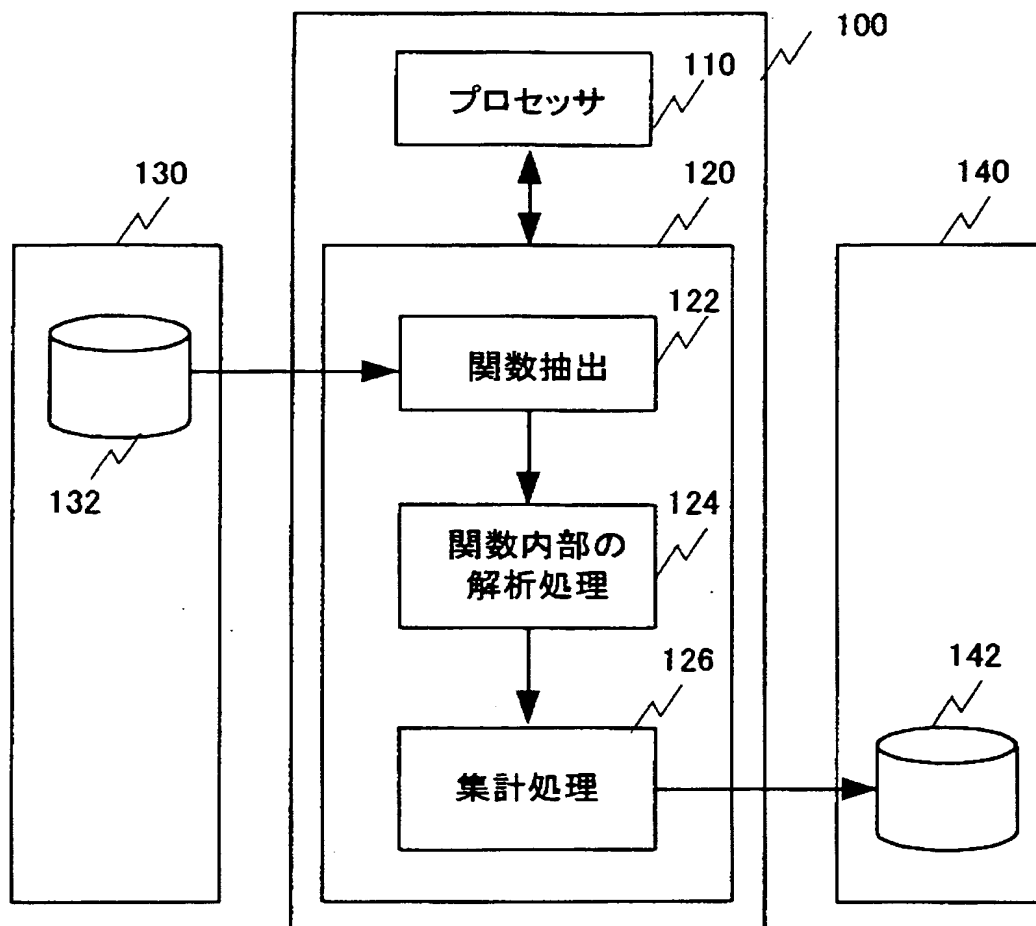
マッピング処理結果例を概念的に表で示した模式図である。

【符号の説明】

- 1 0 0 コンピュータ
- 1 1 0 プロセッサ
- 1 2 0 主記憶装置
- 1 2 2 関数抽出部
- 1 2 4 解析処理部
- 1 2 6 集計処理部
- 1 3 0 補助記憶装置
- 1 4 0 補助記憶装置
- 1 3 2 プログラム
- 1 4 2 集計結果情報
- 2 0 3 2 ファイル
- 2 1 2 2 関数抽出部
- 2 3 2 8 マッピング処理部
- 2 3 3 2 ライブラリ
- 2 3 4 2 マッピング処理結果情報

【書類名】 図面

【図 1】



【図 2】

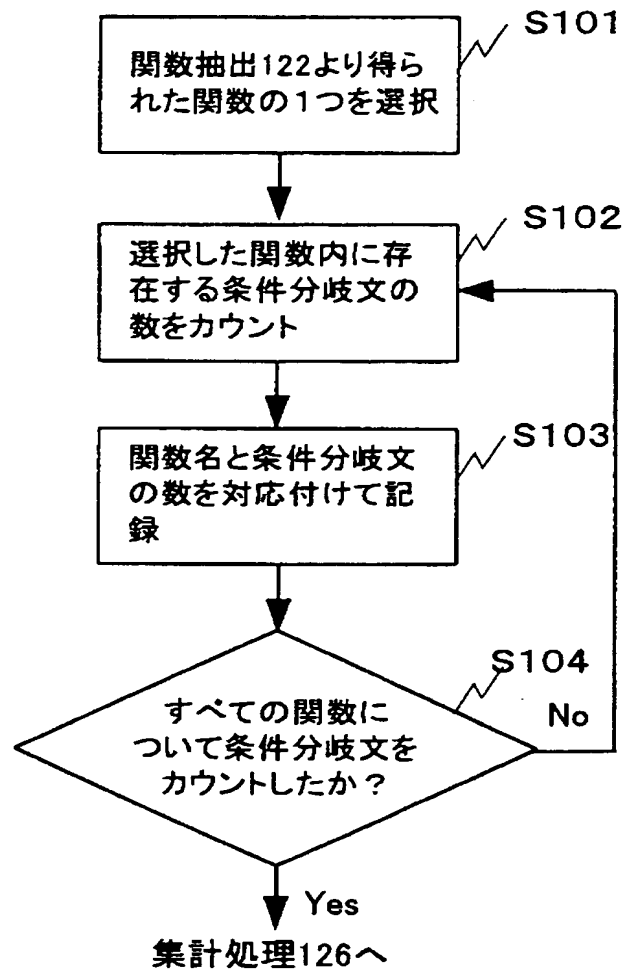
132

```
int      func_A      (AIN1,AIN2)
int      AIN1,AIN2;
{
    211  int      A1, A2;
    int      AO1;
    if (AIN1 == 10) {
        212  if (AIN2 > 20) {
            ...
        }
    } else {
        ...
    }
    213  if (AIN1 > 10) {
        ...
    }
    return (AO1);
}

int      func_B      (BIN1)
int      BIN1;
{
    int      B1, B2;
    int      BO1;
    if (BIN1 > 20) {
        ...
    }
    if (BIN1 > 18 ) {
        ...
    }
    if (BIN1 > 16) {
        ...
    }
    ...
    if (BIN1 > 2) {
        ...
    }
}

void     func_C      ()
{
    ...
}
```

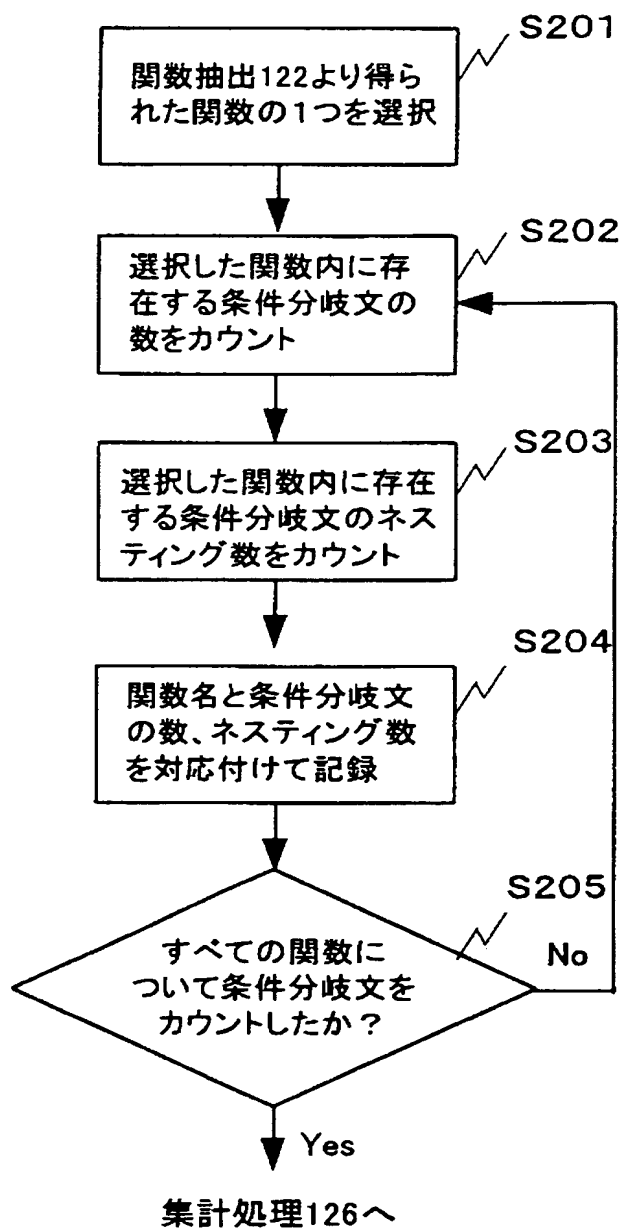
【図 3】



【図 4】

関数名	条件分岐文の数
func_A	3
func_B	10
func_C	0

【図 5】



【図 6】

関数名	条件分岐文		
	数	ネスティング数	
		段数	該当する分岐文の数
func_A	3	0	2
		1	1
func_B	10	0	10
func_C	0	0	0

【図 7】

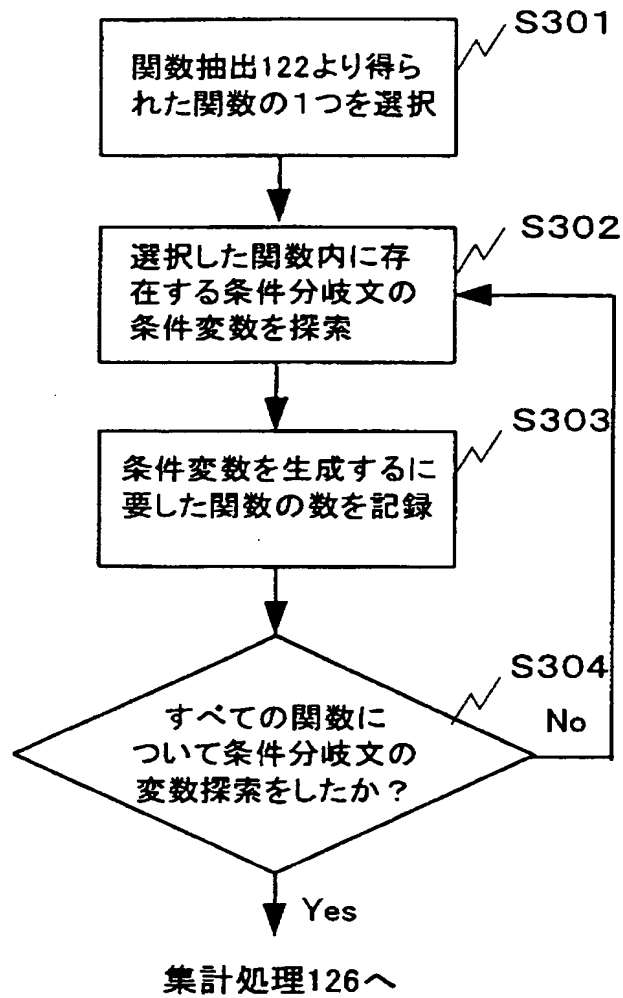
```

132  int      func_A      (AIN1,AIN2)
      int      AIN1,AIN2;
      {
211      int      A1, A2;
      int      AO1;
      if (AIN1 == 10) {
212          if (AIN2 > 20) {
              ...
          }
      } else {
          ...
      }
213      if (AIN1 > 10) {
          ...
      }
      return (AO1);
      }

void      func_D      ()
      {
      int      D1, D2, D3;
711      D2 = func_A1 ( D1 );
712      AIN1 = func_A2 ( D2 );
      ...
      D3 = func_A ( AIN1, D2 );
      }

```


【図 8】



【図 9】

関数名	条件分岐文		
	数	変数生成	
		生成に要する関数	変数の数
func_A	3	2	1
		0	1
func_D	0	0	0

【図 10】

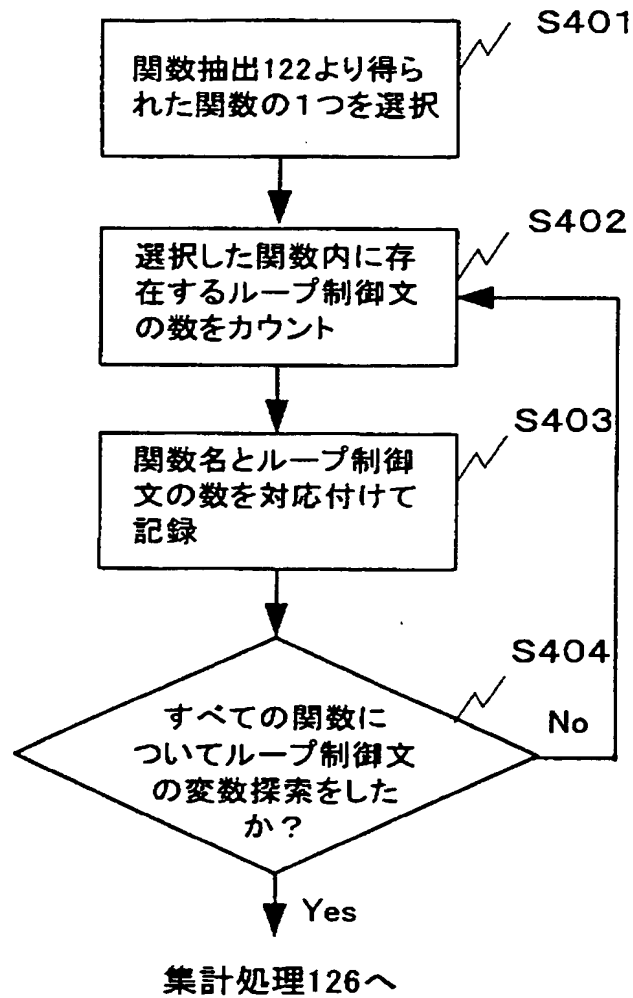
132

```
int      func_LA      (AIN1,AIN2)
int
{
    int      ij;
    int      A1, A2;
    int      AO1;
    1011  for (i = 0; i < AIN1; ++i) {
    1012      for (j = 0; j < AIN2; ++j) {
        ...
        }
    1013  }
    for (i = 0; i < 10; ++i) {
        ...
    }
    return (AO1);
}

int      func_LB      (BIN1)
int
{
    int      i;
    int      B1, B2;
    int      BO1;
    for (i = 0; i < BIN1; ++i) {
        ...
    }
    for (i = 0; i < BIN1; ++i) {
        ...
    }
    for (i = 0; i < BIN1; ++i) {
        ...
    }
    ...
    for (i = 0; i < BIN1; ++i) {
        ...
    }
}

void      func_LC      ()
{
    ...
}
```

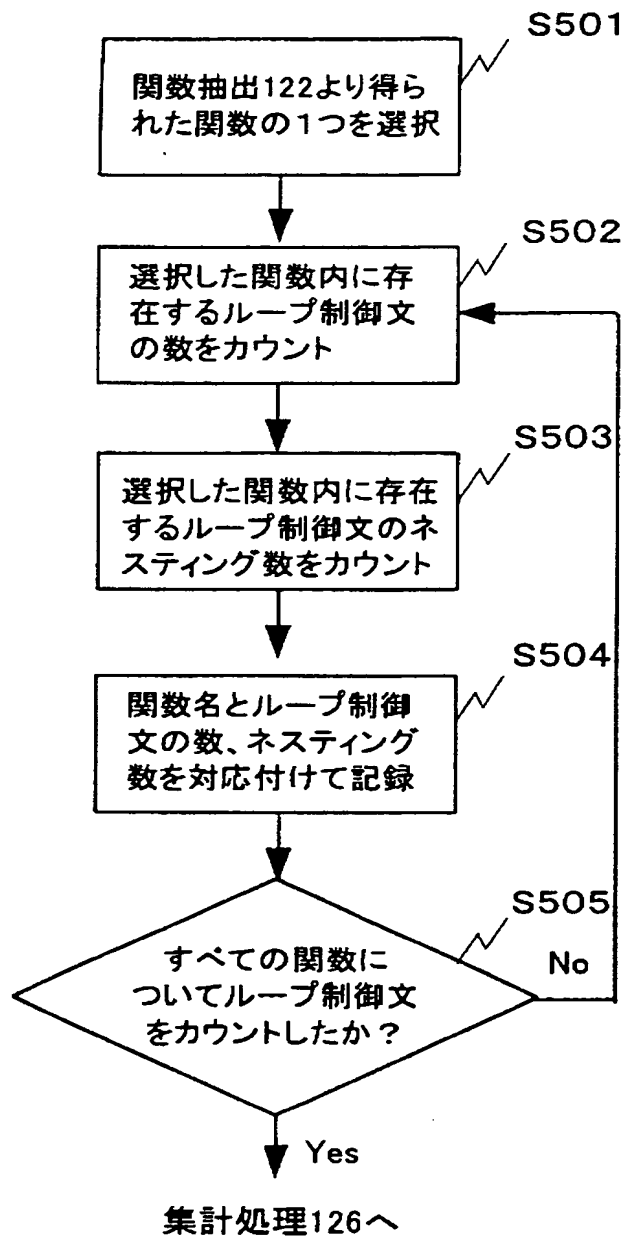
【図 11】



【図 12】

関数名	ループ制御文の数
func_LA	3
func_LB	10
func_LC	0

【図 13】



【図 14】

関数名	ループ制御文		
	数	ネスティング数	
		段数	該当するループ制御文の数
func_LA	3	0	2
		1	1
func_LB	10	0	10
func_LC	0	0	0

【図 15】

132

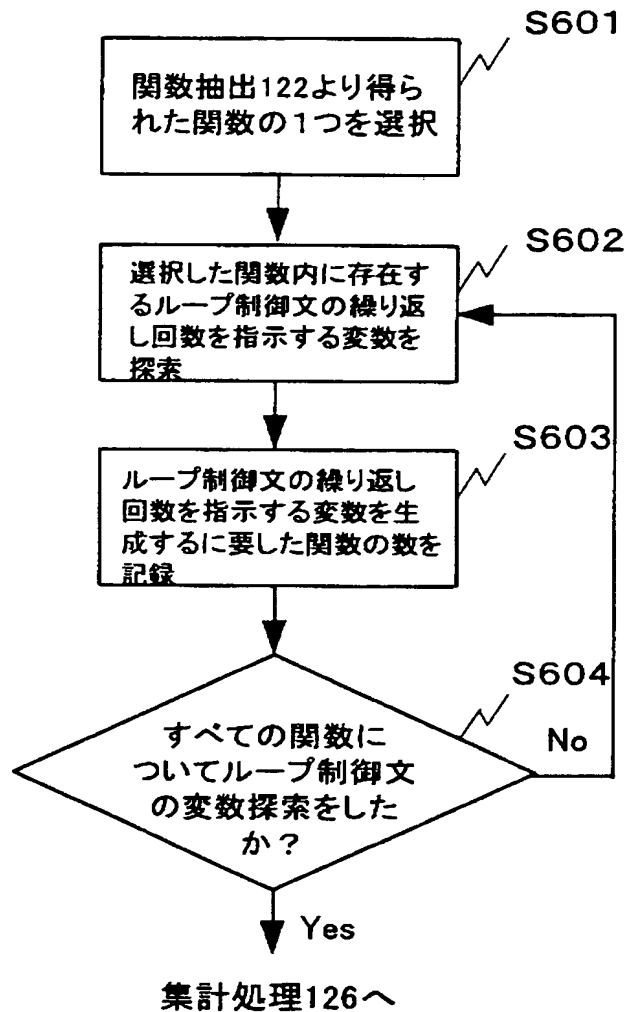
```

int      func_LA  (AIN1,AIN2)
int      AIN1,AIN2;
{
    1011  int      A1, A2;
    int      AO1;
    1012  for (i = 0; i < AIN1; ++i) {
        ...
        }
    1013  }
    for (i = 0; i < 10; ++i) {
        ...
    }
    return (AO1);
}

void      func_LD  ()
{
    1511  int      D1, D2, D3;
    1512  D2 = func_LA1 ( D1 );
    AIN1 = func_LA2 ( D2 );
    ...
    D3 = funcL_A ( AIN1, D2 );
}

```

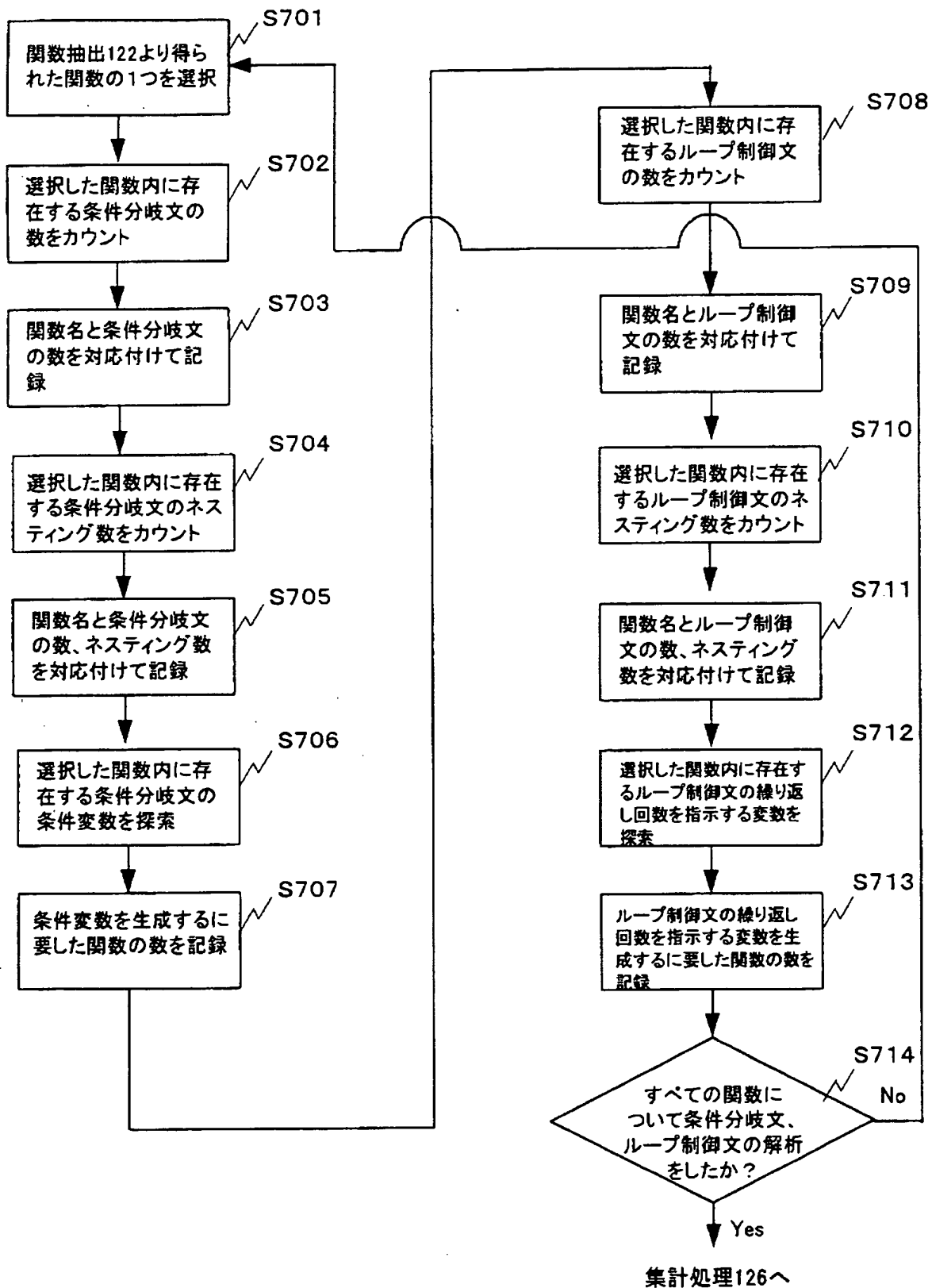
【図 16】



【図 17】

関数名	ループ制御文		
	数	繰り返し回数の変数生成	
		生成に要する関数	変数の数
func_LA	3	2	1
		0	1
func_LD	0	0	0

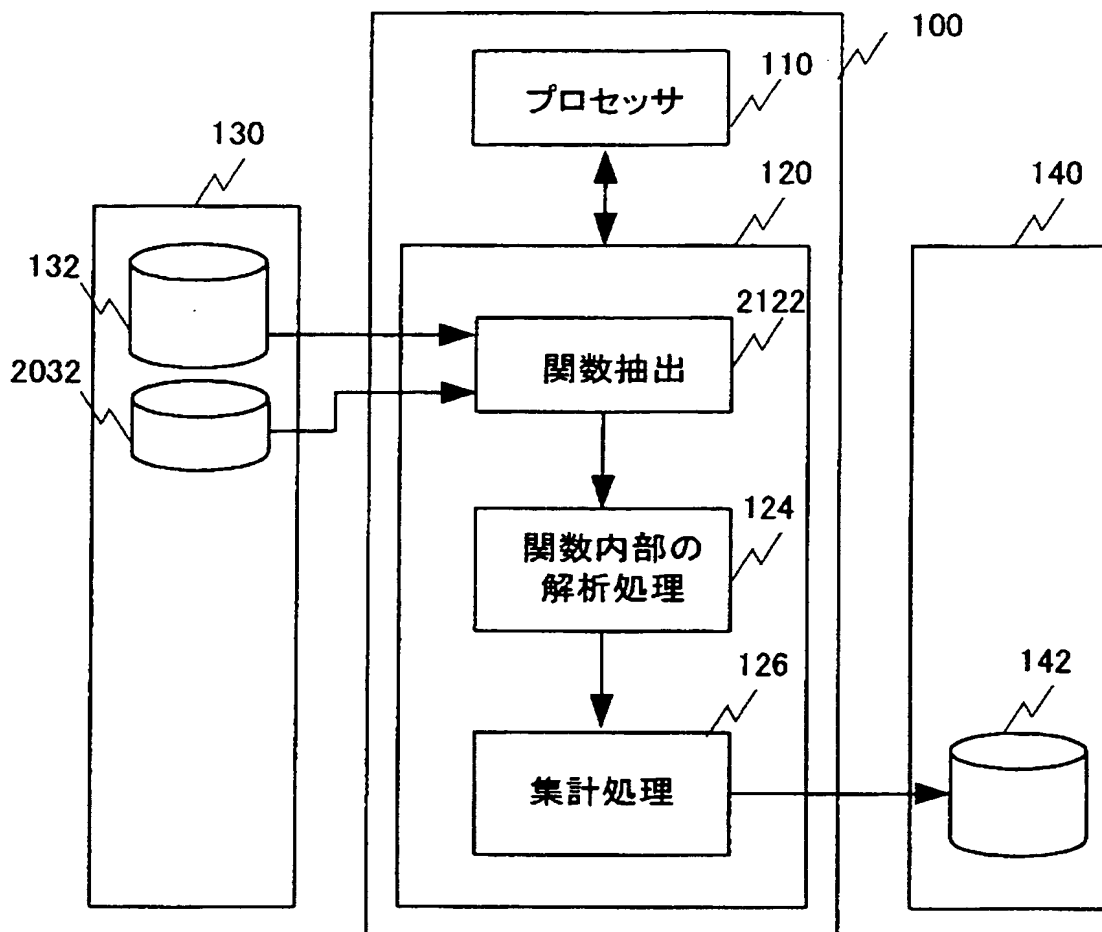
【図 18】



【図 19】

関数名	条件分岐文					ループ制御文				
	数	ネステイング数		変数生成		数	ネステイング数		繰り返し回数 生成に要する関数	変数の数 生成に要する関数
		段数	該当する分岐文の数	生成に要する関数	変数の数		段数	該当するループ制御文の数		
func_A	3	0	2	2	1	0	0	0	0	0
		1	1	0	1					
func_B	10	0	10	0	1	0	0	0	0	0
func_C	0	0	0	0	0	0	0	0	0	0
func_D	0	0	0	0	0	0	0	0	0	0
func_LA	0	0	0	0	0	3	0	2	2	1
							1	1	0	1
func_LB	0	0	0	0	0	10	0	10	0	1
func_LC	0	0	0	0	0	0	0	0	0	0
func_LD	0	0	0	0	0	0	0	0	0	0

【図 20】



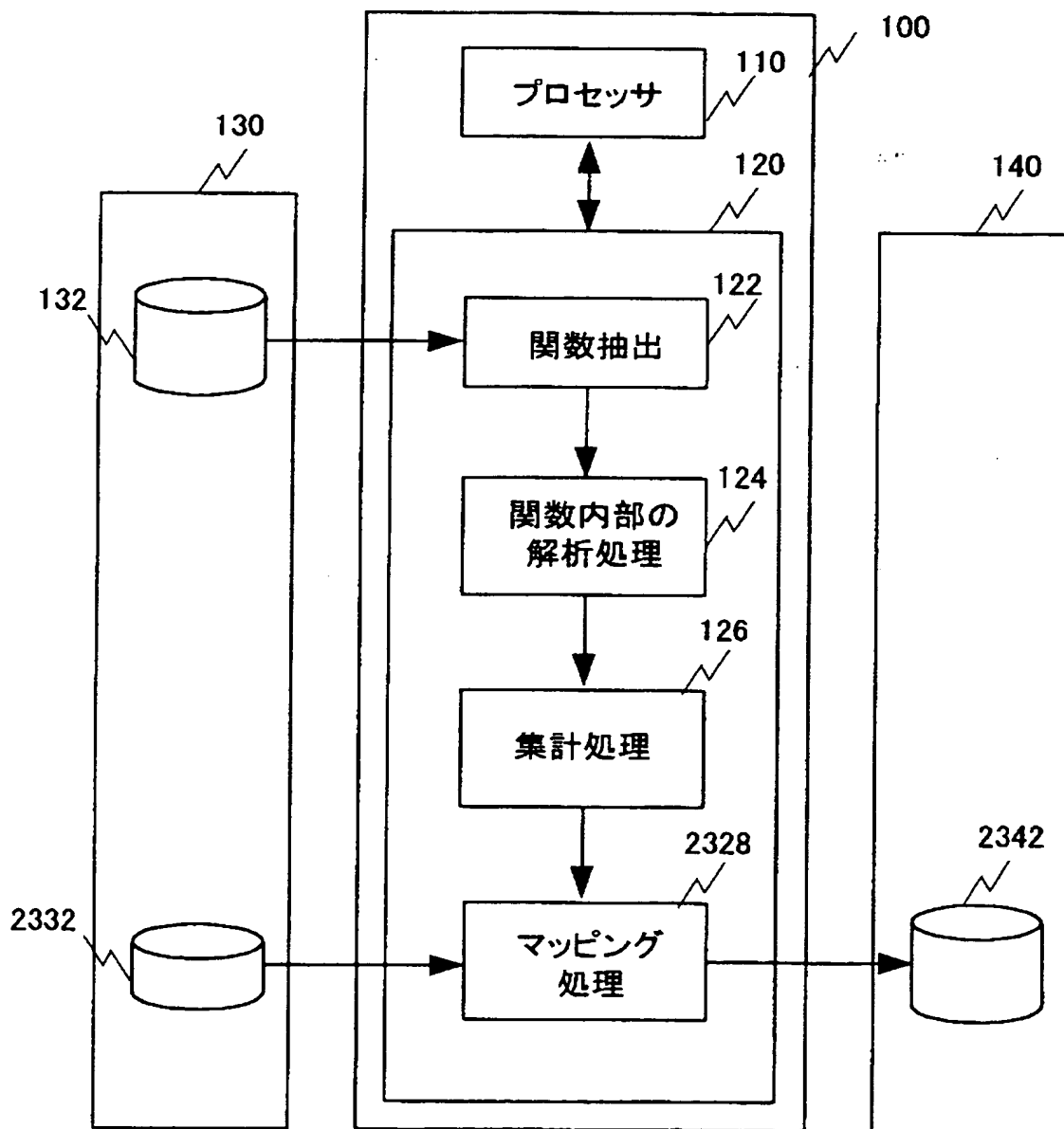
【図 21】

関数名	条件分岐文		
	数	変数生成	
		生成に要する関数	変数の数
func_A + func_D	3	2	1
		0	1

【図 22】

関数名	ループ制御文		
	数	繰り返し回数の変数生成	
		生成に要する関数	変数の数
func_LA + func_LD	3	2	1
		0	1

【図 23】



【図 2 4】

処理ユニット	適した条件分岐文の数
CPU	10以上
DSP	3以上9以下
専用ロジック	2以下

【図 2 5】

関数名	適切な処理ユニット
func_A	DSP
func_B	CPU
func_C	専用ロジック

【図 2 6】

処理ユニット	適したループ制御文の数
CPU	3以上9以下
DSP	10以上
専用ロジック	2以下

【図 2 7】

関数名	適切な処理ユニット
func_LA	CPU
func_LB	DSP
func_LC	専用ロジック

【図 28】

関数名	通した条件分岐文					通したループ制御文			
	数	ネスティング数		変数生成		数	ネスティング数		変数の数
		段数	該当する分岐文の数	生成に要する関数	変数の数		段数	該当するループ制御文の数	
CPU	10以上	0	1以上	0	1以上	3以上9以下	0	3以上9以下	3以上9以下
		1	1以上	1	1以上		1	0	0
		2以上	1以上	2以上	1以上		2以上	0	0
DSP	3以上9以下	0	3以上9以下	0	3以上9以下	10以上	0	1以上	2以下
		1	3以下	1	3以下		1	1以上	2以下
		2以上	1以下	2以上	1以下		2以上	1以上	2以下
専用ロジック	2以下	0	3以下	0	3以下	2以下	0	3以下	3以下

【図 29】

関数名	適した条件分岐文					適したループ制御文			
	数	ネスティング数		変数生成		ネスティング数		繰り返し回数の変数生成	
		段数	該当する分岐文の数	生成に要する関数	変数の数	段数	該当するループ制御文の数	生成に要する関数	変数の数
CPU	K=6	K=1	1以上	K=1	1以上	K=1	3以上9以下	K=1	3以上9以下
		K=2	1以上	K=2	1以上	K=1	0	K=1	0
		K=3	1以上	K=3	1以上	K=1	0	K=1	0
DSP	K=6	K=1	3以上9以下	K=1	3以上9以下	K=1	1以上	K=1	2以下
		K=2	3以下	K=2	3以下	K=2	1以上	K=2	2以下
		K=2	1以下	K=2	1以下	K=3	1以上	K=3	2以下
専用レジック	K=6	K=1	3以下	K=1	3以下	K=2	3以下	K=2	3以下

CPU=3+4+2+2=11

DSP=6+4+4+6=20

専用レジック=1+6+2+2=11

【図 3 0】

関数名	適切な処理ユニット
func_A	DSP
func_B	CPU
func_C	専用ロジック
func_LA	CPU
func_LB	DSP
func_LC	専用ロジック

【書類名】 要約書

【要約】

【課題】 システム L S I に内蔵されている異なったアーキテクチャを持つ処理ユニットにシステムが担う機能を適切に割り当てる。

【解決手段】 システムの機能を高級言語で記述したプログラム内のそれぞれの関数について、その内部に記述された条件分岐文やループ制御文の数、条件分岐文やループ制御文のネスティング数、条件分岐文の条件やループ制御文の繰り返し回数の生成に要した関数の数等を計数する解析手段 1 2 4 を備える。

【選択図】 図 1

特願 2 0 0 3 - 0 5 1 0 4 6

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 5 8 2 1]

1 . 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社